UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/801,080	03/07/2001	Natalino Giorgio Busa	NL000133	5082

24737 7590 05/22/2008
PHILIPS INTELLECTUAL PROPERTY & STANDARDS
P.O. BOX 3001
BRIARCLIFF MANOR, NY 10510

EXAMINER	
PAN, DANIEL H	
ART UNIT	PAPER NUMBER
2183	
MAIL DATE	DELIVERY MODE
05/22/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 09/801,080
Filing Date: March 07, 2001
Appellant(s): BUSA ET AL.

Gregory L. Thorne
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 12/06/06 appealing from the Office action mailed 03/02/06.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

4,876,643	McNeill et al.	10-1989
6,266,766	O'Connor	07-2001

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claim 1-5 are rejected under 35 U.S.C. 102(b) as being anticipated by McNeill et al. (4,876,643).

As to claims 1, McNeill taught a data processing device, comprising at least:
a) a master controller (210), a first functional unit including a slave controller (212),
b) a second functional unit (see the disk manager 16, 22 in fig.1), and
c) a common memory means (214) shared by the first and second functional units (see common memory 214 shared by first and second units, see first unit [212] connected

to share memory 214 in fig.2, see also data can be read from second unit [22] through interface 216 into the snared memory 214); the device being programmed for executing an instruction by the first functional unit (see the programmed slave processor 212 in co1.6, lines 55-61), execution of instruction involving input/output operations by the first functional unit (see the disk interface as input/output in co1.7, lines 36-40), the execution involves at least one of output data (see the search pattern match as output data, see also col.6, lines 65-68, co1.7, lines 20-35) of the first functional unit being processed by the second functional unit during execution of instruction, and the input data to the first functional unit being generated by the second functional unit during execution of the instruction (see the indexed files in co1.3, lines 15-21, see how the master mange communication between the input streams in col.6, lines 39-54).

As to claim 2, McNeill also included relative latency (see wait time in co1.7, lines 5-20).

As to claims 3,5, McNeill also included a halt means (see the wait for next record and list in col.7, lines 2-35).

As to claim 4, McNeill taught a method of operating a data processing device comprising:

a) a master controller for controlling operation of the data processing device, a first functional unit, which includes a slave controller [212] , the first functional unit being arranged for executing instructions of a first type corresponding to operations having a relatively long latency (see search process in col.4, lines 35-48, see also the large number of concurrent searches as a long latency in col.3, lines 35-30);

b) a second functional unit [16,22] capable of executing instructions of a second type corresponding to operations having a relatively short latency (see extracting by disk manager 16 and updating by database 22 as a short latency in col.4, lines 48-50 and col.5, lines 29-38), wherein the first functional unit during execution of an instruction of the-first type receives input data [search key or the search record] (see col.4, lines 45-52) and provides output data [matched pattern or search record] , and execution involves at least one of output data [search record] of the first functional unit [slave 212] being processed by the second functional unit [disk manger 16] during execution of instruction (instruction not explicitly shown, but it must have a search instruction or a search command), and input data (data field stored in disk [22] and provided to search slave 212, col.4, lines 54- fig.4a) to the first functional unit [slave 212] being generated by the second functional unit [22] during execution of the instruction (see database 22 provided the search fields to slave 212 for searching in col.4, lines 55-68) .

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

Claims 1-5 are rejected under 35 U.S.C. 102(e) as being anticipated by O'Connor (6,266,766).

As to claim 1, O'Connor discloses a data processing device comprising

a) a master controller [MUX], a first function unit including a slave controller (see fig.1 10 controlled by MUX),

b) a second function unit (see fig.1 [20]), and a common memory [30] shared by the first and second functional units (see fig. 1 [30]), the data processing device being programmed for executing an instruction [ADD instruction] (col.2, lines 37-43), execution of the instruction involving I/O operations by the first functional unit (see output result and bypass result as input and output operations), wherein the execution involves at least one of output data of the first functional unit being processed by the second function unit in the of execution of the instruction and input the input data to the first function unit being generated by the second functional unit during execution of said instruction. (see col. 1, lines 16-17 and lines 28- 32, see the execution of instructions), the first execution unit outputs data processed by the second function unit and data input to the first function unit by the second function unit (see the feedback data path 16 from execution 10 to the input of execution 20 in fig.1) during execution because the data is then executed by the execution units.

As to claim 2, O'Connor teaches the first function unit is arranged for processing instructions of a first type corresponding to operations having a relatively large latency and the second function unit is arranged for processing instructions of a second type corresponding to operations having a relatively small latency. O'Connor discloses each of the execution units take varying amounts of time to complete (see col. 1, lines 54-61). Therefore, the first functional unit takes longer than the second. And, the first function unit processes operations of a relatively large delay and the second function unit of a relatively small delay.

As to claim 3, O'Connor discloses halt means controllable by the master controller for suspending operation of the first functional unit. The master controller is used for implementing interlocks, which is halting or suspending execution if the data is not ready. Therefore, the first function unit is halted when the data is not.

As to claim 4, O'Connor discloses a method of operating a data processing device, comprising'. a) a master controller for controlling operation of the data processing device (e.g. see col. 1, lines 32-53, see the interlocks when data is no ready). An interlock is needed if the data not been executed and ready for bypassing. Therefore, interlocking exists in the processor and there must be a master controller to control the interlocks. b) a first function unit, which includes a slave controller (see (figure 1, [10] for the first unit), the first function unit [10] being arranged for executing instructions of a first type corresponding to operations having a relatively long latency; (see col. 1, lines 16-17 and lines 28-32, see the execution of instructions. Col. 2, lines 9-10 and figure 2 show that there is a separate controller for controlling the bypass functionality. This is the slave controller. The execution units take varying amounts of time to complete (see col. 1, lines 54-61, see different types of instructions). Therefore, the first function unit takes longer than the second. And, the first function unit processes operations of a relatively large delay. c) a second function unit capable of executing instructions of a second type corresponding to operations having a relatively short delay (see fig.1, [20]), wherein the first function unit during execution of an instruction of the first type receives input data and provides output data, and the execution involves at least one of: output data of the first functional unit being processed by the second functional unit in the midst of execution of said instruction and input data to the first functional unit being generated by the second functional unit in execution of the instruction. The first function unit has a relatively long delay compared to the second unit so the second function unit has a relatively shod latency. The second function unit provides input data for the first function unit and receives output data from the first unit (see fig.1). This is done during execution because the data is then executed by the execution units. As to the language "in the midst of" , it is read as I the close range of. Therefore, a functional unit operating in the close range of execution by other function unit is only operating in time proximate to the other function unit and encompasses all type of processing, such as sequential, concurrent, and other processing.

O'Connor teaches the second execution unit executes data directly given from the first execution unit, the operation of the second execution unit is in close range to the execution of the first execution unit and therefore, operates in the midst of the execution of the first unit.

As to claim 5, O'Connor also discloses his master controller temporarily suspends operation of the first functional unit during execution of instructions of the first type. The master controller is used for implementing interlocks, halting, or suspending execution, if the data is not ready. The first functional unit is halted when data is not ready.

(10) Response to Argument

In the remarks, appellant argued that:

a) O Connor's one execution unit waits for results from another execution unit in order for the first execution unit to begin execution. The execution do not operate concurrently to achieve execution of an instruction (see Appellant's Brief, Page 6);

b) The rejection based on McNeill ignores the basic distinction between an instruction and a computing task. An instruction is a well-defined operation that takes a known number of clock cycles of a functional unit. A computing task, on the other hand, is performed by executing a series of instructions in sequence. The time required to perform the computing task is often indeterminate (see appellant's Brief Page 6).

As to a) above, examiner would like to point out that nowhere does appellant's claim recite "concurrently execution." applicant is reminded that unclaimed features cannot be used to overcome the prior art (e.g. see CCPA In re Lundenberg & Zuschlag, 113, USPQ 530, 534 (1957)). Nevertheless, even if one execution unit had to wait for another execution unit, it did not necessarily mean that it was not a concurrent processing. A concurrent processing could be two execution units run independently and dependently at respective points of time, and produced results simultaneously. O'Connor taught an ADD instruction had to wait for a register result from a LOAD instruction (col.1, lines 22-36). However, O'Connor also taught a bypass used to send result (to be placed in a register) to the waiting execution unit at the same time the result is sent to the register for not having to wait and significantly speeding-up operations (see col1, lines 46-53). Furthermore, O'Connor taught a plural multicycle execution units in pipelined implementation (see co1.3, lines 54-59). Therefore, it must have concurrent processing. See also execution results being returned in out-of - order or parallel fashion in col.7, lines 37-46. Therefore, the execution of O'Connor did operate concurrently to achieve execution of an instruction.

As to b) above, nowhere does applicant specification or claim define the distinction between “instruction” and “task”. Nevertheless, examiner holds that a task has to be in the form of instruction in order to carry out the execution. A task can be single instruction task or multi-instructions task. An instruction itself is a task. For example, instruction MOVE R1, Mem is a task (Moving content of memory to register R1). Furthermore, appellant’s claimed scope is not directed to the distinction between an instruction and a task. In addition, McNeill taught a master processor received a search request and assigned execution of the search request to a slave processor (col.4, lines 35-37). A search request is a command. And, therefore, a search request is an instruction.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/Daniel Pan/

Primary Examiner, Art Unit 2183

Conferees:

/Kevin L Ellis/
Acting SPE of Art Unit 2187

/Eddie P Chan/
Supervisory Patent Examiner, Art Unit 2183